


```
FFFFFFFFF  IIIII  LL  EEEEEEEEE  CCCCCC  MM  MM  DDDDDDD  SSSSSSS
FFFFFFFFF  IIIII  LL  EEEEEEEEE  CCCCCC  MM  MM  DDDDDDD  SSSSSSS
FF        II    LL  EEE         CC      MMM  MMM  DD      DD  SS
FF        II    LL  EEE         CC      MMM  MMM  DD      DD  SS
FF        II    LL  EEE         CC      MM   MM   DD      DD  SS
FF        II    LL  EEEEEEE     CC      MM   MM   DD      DD  SSSSSS
FFFFFFFFF  II    LL  EEE         CC      MM   MM   DD      DD  SSSSSS
FFFFFFFFF  II    LL  EEE         CC      MM   MM   DD      DD  SS
FF        II    LL  EEE         CC      MM   MM   DD      DD  SS
FF        II    LL  EEE         CC      MM   MM   DD      DD  SS
FF        II    LL  EEE         CC      MM   MM   DD      DD  SS
FF        IIIII  LLLLLLLLL  EEEEEEEEE  CCCCCC  MM  MM  DDDDDDD  SSSSSSS
FF        IIIII  LLLLLLLLL  EEEEEEEEE  CCCCCC  MM  MM  DDDDDDD  SSSSSSS

LL        IIIII  SSSSSSS
LL        IIIII  SSSSSSS
LL        II    SS
LL        II    SS
LL        II    SS
LL        II    SS
LL        II    SSSSSS
LL        II    SSSSSS
LL        II    SS
LL        II    SS
LL        II    SS
LL        IIIII  SSSSSSS
LLLLLLLLL  IIIII  SSSSSSS
LLLLLLLLL  IIIII  SSSSSSS
```

```
0001 0 title 'FILECMDS File-Oriented Phone Commands'
0002 0 module filecmds (
0003 1 ident='V04-000') = begin
0004 1
0005 1
0006 1 *****
0007 1 *
0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0010 1 * ALL RIGHTS RESERVED.
0011 1 *
0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0017 1 * TRANSFERRED.
0018 1 *
0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0021 1 * CORPORATION.
0022 1 *
0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0025 1 *
0026 1 *****
0027 1
0028 1
0029 1
0030 1 ++
0031 1 Facility: VAX/VMS Telephone Facility, File-Oriented Phone Commands
0032 1
0033 1 Abstract: This module handles the following user commands:
0034 1 FACSIMILE Include file contents in conversation.
0035 1 TRANSCRIBE Make a transcript of conversation.
0036 1
0037 1
0038 1 Environment:
0039 1
0040 1 Author: Paul C. Anagnostopoulos, Creation Date: 6 November 1980
0041 1
0042 1 Modified By:
0043 1
0044 1 : Version
0045 1 01 -
0046 1 --
```



```
48 0047 1 %sbttl 'Module Declarations'
49 0048 1
50 0049 1 Libraries and Requires:
51 0050 1
52 0051 1
53 0052 1 library 'sys$library:starlet.l32';
54 0053 1 require 'phonereq';
55 0382 1
56 0383 1
57 0384 1 Table of Contents:
58 0385 1
59 0386 1
60 0387 1 forward routine
61 0388 1     phn$facsimile_cmd: novalue,
62 0389 1     phn$facsimile2: novalue,
63 0390 1     phn$transcribe_cmd: novalue;
64 0391 1
65 0392 1
66 0393 1 The following macro is used in PHN$TRANSCRIBE_CMD:
67 0394 1
68 0395 1
69 M 0396 1 macro transcribe(length,address) =
70 M 0397 1     if dec(line_counter) lss 0 then
71 M 0398 1         page_headings();
72 M 0399 1         trans_rab[rab$w_rsz] = length;
73 M 0400 1         trans_rab[rab$l_rbf] = address;
74 M 0401 1         status = $put(rab=trans_rab);
75 M 0402 1         check(.status);
76 0403 1 %;
77 0404 1
78 0405 1
79 0406 1 External References:
80 0407 1
81 0408 1
82 0409 1 external routine
83 0410 1     phn$conversing,
84 0411 1     phn$fresh_screen,
85 0412 1     phn$inform,
86 0413 1     phn$kill_ctl,
87 0414 1     phn$queue_smb,
88 0415 1     phn$talk;
89 0416 1
90 0417 1
91 0418 1 Own Variables:
92 0419 1
93 0420 1
94 0421 1 own
95 P 0422 1     facs_fab: $fab( fac=get,      ! RMS FAB for facsimile file.
96 0423 1         shr=get),
97 0424 1     facs_buffer: block[79,byte], ! Buffer for getting file records.
98 P 0425 1     facs_rab: $rab( fab=facs_fab, ! RMS RAB for facsimile file.
99 0426 1         rac=seq,
100 P 0427 1         rop=loc,
101 P 0428 1         ubf=facs_buffer,
102 0429 1         usz=79);
```

```
104 0430 1 %sbttl 'PHN$FACSIMILE_CMD - Include File in Conversation'
105 0431 1 ++
106 0432 1 Functional Description:
107 0433 1 This routine handles the FACSIMILE command, which is used to
108 0434 1 include the contents of a file into the conversation.
109 0435 1
110 0436 1 Formal Parameters:
111 0437 1 facs_spec Address of descriptor of file specification.
112 0438 1
113 0439 1 Implicit Inputs:
114 0440 1 global data
115 0441 1
116 0442 1 Implicit Outputs:
117 0443 1 global data
118 0444 1
119 0445 1 Returned Value:
120 0446 1 none
121 0447 1
122 0448 1 Side Effects:
123 0449 1
124 0450 1 --
125 0451 1
126 0452 1
127 0453 2 global routine phn$facsimile_cmd(facs_spec): novalue = begin
128 0454 2
129 0455 2 bind
130 0456 2 spec_dsc = .facs_spec: descriptor;
131 0457 2
132 0458 2 local
133 0459 2 status: long;
134 0460 2
135 0461 2
136 0462 2 ! Check to make sure we are currently talking to someone. Otherwise facsing
137 0463 2 ! makes no sense.
138 0464 2
139 0465 2 if not phn$conversing() then (
140 0466 2 phn$inform(phn$_notconv);
141 0467 2 return;
142 0468 2 );
143 0469 2
144 0470 2 ! Now we have to open the file we are going to facs. It is also necessary to
145 0471 2 ! connect the RAB so we can read records. If either of these operations fails,
146 0472 2 ! tell the user why.
147 0473 2
148 0474 2 facs_fab[fab$b_fns] = .spec_dsc[len];
149 0475 2 facs_fab[fab$l_fna] = .spec_dsc[ptr];
150 0476 2 status = $open(fab=facs_fab);
151 0477 2 if .status nequ rms$normal then (
152 0478 2 phn$inform(.status);
153 0479 2 return;
154 0480 2 );
155 0481 2 status = $connect(rab=facs_rab);
156 0482 2 if .status nequ rms$normal then (
157 0483 2 $close(fab=facs_fab);
158 0484 2 phn$inform(.status);
159 0485 2 return;
160 0486 2 );
```

```
161 0487 2
162 0488 2 ! Tell the user how to cancel the operation.
163 0489 2
164 0490 2 phn$inform(phn$_facscan);
165 0491 2
166 0492 2 ! Set the "facsimile in progress" flag so it can be cancelled if necessary.
167 0493 2
168 0494 2 phn$gv_facsimile = true;
169 0495 2
170 0496 2 ! Talk a carriage return so the file will start on a new line. Queue a
171 0497 2 ! steering message that will force us to facs the first record.
172 0498 2
173 0499 2 phn$talk(describe(%char(ret)));
174 0500 2 phn$queue_smb(smb__facsimile2);
175 0501 2 return;
176 0502 2
177 0503 1 end;
```

```
.TITLE FILECMDS FILECMDS File-Oriented Phone Commands
.IDENT \V04-000\
```

```
.PSECT SPLITS,NOWRT,NOEXE,2
```

```
0D 00000 P.AAB: .ASCII <13>
00000001 00001 .BLKB 3
00000000 00004 P.AAA: .LONG 1
00000000 00008 .ADDRESS P.AAB
```

```
.PSECT $OWNS$,NOEXE,2
```

```
03 00000 FACS_FAB:
50 00001 .BYTE 3
0000 00002 .BYTE 80
00000000 00004 .WORD 0
00000000 00008 .LONG 0
00000000 0000C .LONG 0
00000000 00010 .LONG 0
0000 00014 .WORD 0
02 00016 .BYTE 2
02 00017 .BYTE 2
00000000 00018 .LONG 0
00 0001C .BYTE 0
00 0001D .BYTE 0
00 0001E .BYTE 0
02 0001F .BYTE 2
00000000 00020 .LONG 0
00000000 00024 .LONG 0
00000000 00028 .LONG 0
00000000 0002C .LONG 0
00000000 00030 .LONG 0
00 00034 .BYTE 0
00 00035 .BYTE 0
0000 00036 .WORD 0
00000000 00038 .LONG 0
0000 0003C .WORD 0
```



```

00 0003E .BYTE 0
00 0003F .BYTE 0
00000000 00040 .LONG 0
00000000 00044 .LONG 0
0000 00048 .WORD 0
00 0004A .BYTE 0
00 0004B .BYTE 0
00000000 0004C .LONG 0
00050 FACS_BUFFER:
0009F .BLKB 79
01 000A0 FACS_RAB:
0009F .BLKB 1
44 000A1 .BYTE 1
0000 000A2 .BYTE 68
00010000 000A4 .WORD 0
00000000 000A8 .LONG 65536
00000000 000AC .LONG 0
0000# 000B0 .LONG 0
0000 000B6 .WORD 0[3]
00000000 000B8 .WORD 0
0000 000BC .LONG 0
00 000BE .WORD 0
00 000BF .BYTE 0
004F 000C0 .BYTE 0
0000 000C2 .WORD 79
00000000 000C4 .WORD 0
00000000 000C8 .ADDRESS FACS_BUFFER
00000000 000CC .LONG 0
00000000 000D0 .LONG 0
00 000D4 .LONG 0
00 000D5 .BYTE 0
00 000D6 .BYTE 0
00 000D7 .BYTE 0
00000000 000D8 .LONG 0
00000000 000DC .ADDRESS FACS_FAB
00000000 000E0 .LONG 0

```

```

.EXTRN PHNS_OK, PHNS_ANSWERED
.EXTRN PHNS_BUSYCALL, PHNS_CANCALL
.EXTRN PHNS_CANTREACH, PHNS_CONFCALL
.EXTRN PHNS_DEAD, PHNS_DECNETLINK
.EXTRN PHNS_DIRCAN, PHNS_FACSCAN
.EXTRN PHNS_HELPCAN, PHNS_HUNGUP
.EXTRN PHNS_JUSTRANG, PHNS_LOGGEDOFF
.EXTRN PHNS_REJECTED, PHNS_RING
.EXTRN PHNS_REJECTJUNK
.EXTRN PHNS_SENDINGMAIL
.EXTRN PHNS_BADCMD, PHNS_BADHELP
.EXTRN PHNS_BADMAILCMD
.EXTRN PHNS_BADSMB, PHNS_BADSPEC
.EXTRN PHNS_HELPMISSING
.EXTRN PHNS_IVREDUNANS
.EXTRN PHNS_IVREDUNCALL
.EXTRN PHNS_LINKERROR, PHNS_NEEDUSER
.EXTRN PHNS_NOCALL, PHNS_NOROLDS
.EXTRN PHNS_NOPORTS, PHNS_NOPRIV

```

				000C 00000	.EXTRN	PHNS_NOPROC, PHNS_NOTCONV	
					.EXTRN	PHNS_ONLYNODE, PHNS_PHONEBUSY	
					.EXTRN	PHNS_REMOTEERROR	
					.EXTRN	PHNS_TARGTERM, PHNS_UNPLUGGED	
					.EXTRN	PHNS_BADTERM, PHNS_SHAREDMBX	
					.EXTRN	PHNS_INPUTTERM, PHNSGQ_NODE_NAME	
					.EXTRN	PHNSGQ_SWITCH_HOOK	
					.EXTRN	PHNSGL_VIEWPORT_SIZE	
					.EXTRN	PHNSGB_SCROLL, PHNSGQ_PUBHEAD	
					.EXTRN	PHNSGB_FLAGS, PHNSCONVERSING	
					.EXTRN	PHNSFRESH_SCREEN	
					.EXTRN	PHNSINFORM, PHNSKILL_CTL	
					.EXTRN	PHNSQUEUE_SMB, PHNSTALK	
					.EXTRN	SYSSOPEN, SYSSCONNECT	
					.EXTRN	SYSSCLOSE	
					.PSECT	\$CODE\$,NOWRT,2	
					.ENTRY	PHNSFACSIMILE_CMD, Save R2,R3	: 0453
					MOVAB	FACS_FAB, R3	:
					MOVL	FACS_SPEC, R2	: 0456
					CALLS	#0, PHNSCONVERSING	: 0465
					BLBS	R0, 1\$:
					PUSHL	#PHNS_NOTCONV	: 0466
					BRB	3\$:
					MOVB	(R2), FACS_FAB+52	: 0474
					MOVL	4(R2), FACS_FAB+44	: 0475
					PUSHL	R3	: 0476
					CALLS	#1, SYSSOPEN	:
					MOVL	R0, STATUS	:
					CMPL	STATUS, #65537	: 0477
					BNEQ	2\$:
					PUSHAB	FACS_RAB	: 0481
					CALLS	#1, SYSSCONNECT	:
					MOVL	R0, STATUS	:
					CMPL	STATUS, #65537	: 0482
					BEQL	4\$:
					PUSHL	R3	: 0483
					CALLS	#1, SYSSCLOSE	:
					PUSHL	STATUS	: 0484
					CALLS	#1, PHNSINFORM	:
					RET		: 0482
					PUSHL	#PHNS_FACSCAN	: 0490
					CALLS	#1, PHNSINFORM	:
					BISB2	#8, PHNSGB_FLAGS	: 0494
					PUSHAB	P.AAA	: 0499
					CALLS	#1, PHNSTALK	:
					PUSHL	#16	: 0500
					CALLS	#1, PHNSQUEUE_SMB	:
					RET		: 0503

; Routine Size: 130 bytes, Routine Base: \$CODE\$ + 0000


```
179 0504 1 %sbttl 'PHN$FACSIMILE2 - Facsimile a Record'
180 0505 1 ++
181 0506 1 Functional Description:
182 0507 1 This steering message routine is called to facsimile the next record
183 0508 1 from the file.
184 0509 1
185 0510 1 Formal Parameters:
186 0511 1 none
187 0512 1
188 0513 1 Implicit Inputs:
189 0514 1 global data
190 0515 1
191 0516 1 Implicit Outputs:
192 0517 1 global data
193 0518 1
194 0519 1 Returned Value:
195 0520 1 none
196 0521 1
197 0522 1 Side Effects:
198 0523 1
199 0524 1 --
200 0525 1
201 0526 1
202 0527 2 global routine phn$facsimile2: novalue = begin
203 0528 2
204 0529 2 local
205 0530 2 status: long,
206 0531 2 record_dsc: descriptor;
207 0532 2
208 0533 2
209 0534 2 ! If the facsimile operation has been cancelled, talk a little message to
210 0535 2 ! that effect and bag the operation.
211 0536 2
212 0537 2 if not .phn$gv facsimile then (
213 L 0538 2 phn$taTk(describe(%string('(CANCELLED)',
214 0539 2 %char(ret))));
215 0540 2 $close(fab=facs_fab);
216 0541 2 return;
217 0542 2 );
218 0543 2
219 0544 2 ! Read the next record from the file. There are three cases:
220 0545 2 1. We get a record (even if it is truncated). Build a descriptor
221 0546 2 for it and talk it into the conversation, with a carriage
222 0547 2 return. Queue a steering message for the next record.
223 0548 2
224 0549 2 2. We get end of file. Talk a little message to that effect and
225 0550 2 cancel the operation.
226 0551 2
227 0552 2 3. We get an I/O error. Talk a little message and cancel.
228 0553 2
229 0554 2 status = $get(rab=facs_rab);
230 0555 2
231 0556 2 selectoneu .status of set
232 0557 2 [rms$normal,
233 0558 2 rms$rtb]: (record_dsc[len] = .facs_rab[rab$w_rsz];
234 0559 2 record_dsc[ptr] = .facs_rab[rab$l_rbf];
235 0560 2 phn$taTk(record_dsc);
```

```
.PSECT SPLITS,NOWRT,NOEXE,2
```

Address	Hex Data	Label	Comment
0D 29 44 45 4C 4C 45 43 4E 41 43 28	0000000C	P.AAD:	.ASCII \ (CANCELLED) \ <13>
	00000000	P.AAC:	.LONG 12
	00000000		.ADDRESS P.AAD
	0D 00020	P.AAF:	.ASCII <13>
	00021		.BLKB 3
	00000001	P.AAE:	.LONG 1
	00000000		.ADDRESS P.AAF
0D 29 46 4F 45 28	0002C	P.AAH:	.ASCII \ (EOF) \ <13>
	00032		.BLKB 2
	00000006	P.AAG:	.LONG 6
	00000000		.ADDRESS P.AAH
0D 29 52 4F 52 52 45 20 45 4C 49 46 28	0003C	P.AAJ:	.ASCII \ (FILE ERROR) \ <13>
	00049		.BLKB 3
	0000000D	P.AAI:	.LONG 13
	00000000		.ADDRESS P.AAJ

```

.EXTRN  SYS$GET

```

.PSECT SCODES,NOWRT,2

			000C	00000	.ENTRY	PHN\$FACSIMILE2, Save R2,R3
	53	00000000G	00	9E	00002	MOVAB SYSS\$CLOSE, R3
	52	0000G	CF	9E	00009	MOVAB PHN\$TALK, R2
	5E		08	C2	0000E	SUP ? #8, SP
OF	0000G		03	E0	00011	BBS #3, PHN\$GB_FLAGS, 1\$
		0000'	CF	9F	00017	PUSHAB P.AAC
	62		01	FB	0001B	CALLS #1, PHN\$TALK
		0000'	CF	9F	0001E	PUSHAB FACS FAB
	63		01	FB	00022	CALLS #1, SYSS\$CLOSE
				04	00025	RET
		0000'	CF	9F	00026	1\$: PUSHAB FACS RAB
00000000G	00		01	FB	0002A	CALLS #1, SYSS\$GET
00010001	8F		50	D1	00031	CMPL STATUS, #65537
			09	13	00038	BEQL 2\$
000181A8	8F		50	D1	0003A	CMPL STATUS, #98728
			1F	12	00041	BNEQ 3\$

0527
0537
0539
0540
0537
0554
0557

FILECMDS
V04-000

FILECMDS File-Oriented Phone Commands
PHNSFACSIMILE2 - Facsimile a Record

1 16
16-Sep-1984 02:09:16
14-Sep-1984 12:53:25

VAX-11 BLISS-32 V4.0-742
[PHONE.SRC]FILECMDS.B32;1

Page 9
(4)

04	6E	0000'	CF	B0	00043	2\$:	MOVW	FACS_RAB+34, RECORD_DSC	: 0558
	AE	0000'	CF	D0	00048		MOVL	FACS_RAB+40, RECORD_DSC+4	: 0559
			5E	DD	0004E		PUSHL	SP	: 0560
	62		01	FB	00050		CALLS	#1, PHNSTALK	: 0561
		0000'	CF	9F	00053		PUSHAB	P.AAE	: 0562
	62		01	FB	00057		CALLS	#1, PHNSTALK	: 0563
			10	DD	0005A		PUSHL	#16	: 0564
0000G	CF		01	FB	0005C		CALLS	#1, PHNSQUEUE_SMB	: 0565
				04	00061		RET		: 0566
0001827A	8F		50	D1	00062	3\$:	CMPL	STATUS, #98938	: 0567
			06	12	00069		BNEQ	4\$: 0568
		0000'	CF	9F	0006B		PUSHAB	P.AAG	: 0569
			04	11	0006F		BRB	5\$: 0570
		0000'	CF	9F	00071	4\$:	PUSHAB	P.AAI	: 0571
	62		01	FB	00075	5\$:	CALLS	#1, PHNSTALK	: 0572
		0000'	CF	9F	00078		PUSHAB	FACS_FAB	: 0573
	63		01	FB	0007C		CALLS	#1, SYSSCLOSE	: 0574
0000G	CF		08	8A	0007F		BICB2	#8, PHNSGB_FLAGS	: 0575
				04	00084		RET		: 0576

; Routine Size: 133 bytes, Routine Base: \$CODE\$ + 0082


```
253 0577 1 %sbttl 'PHN$TRANSCRIBE_CMD - Handle TRANSCRIBE Command'
254 0578 1 ++
255 0579 1 Functional Description:
256 0580 1 This routine handles the TRANSCRIBE command, which is used to make a
257 0581 1 transcript of the conversation. We record the conversation that
258 0582 1 has taken place UP TO THIS POINT, but only for people we are currently
259 0583 1 talking to.
260 0584 1
261 0585 1 Formal Parameters:
262 0586 1 trans_spec The address of the descriptor of the file to receive
263 0587 1 the transcript.
264 0588 1
265 0589 1 Implicit Inputs:
266 0590 1 global data
267 0591 1
268 0592 1 Implicit Outputs:
269 0593 1 global data
270 0594 1
271 0595 1 Returned Value:
272 0596 1 none
273 0597 1
274 0598 1 Side Effects:
275 0599 1
276 0600 1 --
277 0601 1
278 0602 1
279 0603 2 global routine phn$transcribe_cmd(trans_spec): novalue = begin
280 0604 2
281 0605 2 bind
282 0606 2 spec_dsc = .trans_spec: descriptor;
283 0607 2 own
284 0608 2 trans_fab: $fab(dnm='.PHN',
285 0609 2 fac=put,
286 0610 2 mrs=132,
287 0611 2 org=seq,
288 0612 2 rat=cr,
289 0613 2 rfm=var,
290 0614 2 shr=nil),
291 0615 2 trans_rab: $rab(fab=trans_fab,
292 0616 2 rac=seq);
293 0617 2
294 0618 2 own ! These variables are OWN because
295 0619 2 page_number: long, ! of up-level referencing.
296 0620 2 line_counter: long;
297 0621 2 literal
298 0622 2 array_size = 20/pub_k_minlines;
299 0623 2 local
300 0624 2 status: long,
301 0625 2 pub_array: vector[array_size,long],
302 0626 2 ctl_array: vector[array_size,long],
303 0627 2 p: ref pub, cur_p: ref pub,
304 0628 2 min_stamp: long,
305 0629 2 i: long, min_i: long,
306 0630 2 c: ref ctl,
307 0631 2 line_buf: block[132,byte];
```



```

00000000 20 000FB .BYTE 32
00000000 00 000FC .LONG 0
00000000 00 00100 .BYTE 0
00000000 00 00101 .BYTE 0
00000000 02 00102 .BYTE 2
00000000 02 00103 .BYTE 2
00000000 00 00104 .LONG 0
00000000 00 00108 .LONG 0
00000000 00 0010C .LONG 0
00000000 00 00110 .LONG 0
00000000 00 00114 .ADDRESS P.AAK
00000000 00 00118 .BYTE 0
00000000 04 00119 .BYTE 4
00000000 0084 0011A .WORD 132
00000000 00 0011C .LONG 0
00000000 00 00120 .WORD 0
00000000 00 00122 .BYTE 0
00000000 00 00123 .BYTE 0
00000000 00 00124 .LONG 0
00000000 00 00128 .LONG 0
00000000 00 0012C .WORD 0
00000000 00 0012E .BYTE 0
00000000 00 0012F .BYTE 0
00000000 00 00130 .LONG 0
01 00134 TRANS_RAB:
00000000 44 00135 .BYTE 1
00000000 0000 00136 .BYTE 68
00000000 00000000 00138 .WORD 0
00000000 00000000 0013C .LONG 0
00000000 00000000 00140 .LONG 0
00000000 00000000 00144 .WORD 0[3]
00000000 0000 0014A .WORD 0
00000000 00000000 0014C .LONG 0
00000000 0000 00150 .WORD 0
00000000 00 00152 .BYTE 0
00000000 00 00153 .BYTE 0
00000000 0000 00154 .WORD 0
00000000 0000 00156 .WORD 0
00000000 00000000 00158 .LONG 0
00000000 00000000 0015C .LONG 0
00000000 00000000 00160 .LONG 0
00000000 00000000 00164 .LONG 0
00000000 00 00168 .BYTE 0
00000000 00 00169 .BYTE 0
00000000 00 0016A .BYTE 0
00000000 00 0016B .BYTE 0
00000000 00000000 0016C .LONG 0
00000000 00000000 00170 .ADDRESS TRANS_FAB
00000000 00000000 00174 .LONG 0
00178 PAGE_NUMBER:
0017C LINE_COUNTER:
.BKLB 4
.BKLB 4
.EXTRN SYSS$PUT, SYSS$FAO

```


.PSECT \$CODE\$,NOWRT,2

007C 00000 PAGE_HEADINGS:

56	FB	AF	9E	00002	WORD	Save R2,R3,R4,R5,R6	0635
55	00000000G	00	9E	00006	MOVAB	PAGE HEADINGS, R6	
54	00000000G	00	9E	0000D	MOVAB	SYSSPUT, R5	
53	0000	CF	9E	00014	MOVAB	LIB\$SIGNAL, R4	
5E	FF78	CE	9E	00019	MOVAB	LINE COUNTER, R3	
7E	84	BF	9A	0001E	MOVZBL	-136(TSP), SP	
04	AE	AE	9E	00022	MOVAB	#132, FAO_BUF	0640
63	08	AE	9E	00022	MOVAB	FAO_BUF+8, FAO_BUF+4	
	FC	37	D0	00027	MOVL	#55, LINE COUNTER	0643
		A3	D5	0002A	TSTL	PAGE_NUMBER	0645
		06	13	0002D	BEQL	1\$	
03		63	F4	0002F	SOBGEQ	LINE COUNTER, 1\$	0646
66		00	FB	00032	CALLS	#0, PAGE HEADINGS	
DA		01	B0	00035	MOVW	#1, TRANS_RAB+34	
EO	A3	CF	9E	00039	MOVAB	P.AAL, TRANS_RAB+40	
	0000	A3	9F	0003F	PUSHAB	TRANS_RAB	
	B8	01	FB	00042	CALLS	#1, SYSSPUT	
65		50	D0	00045	MOVL	R0, STATUS	
52		52	E8	00048	BLBS	STATUS, 2\$	
05		52	DD	0004B	PUSHL	STATUS	
		01	FB	0004D	CALLS	#1, LIB\$SIGNAL	
64		A3	D6	00050	INCL	PAGE_NUMBER	0647
	FC	A3	DD	00053	PUSHL	PAGE_NUMBER	0654
	FC	7E	D4	00056	CLRL	-(SP)	
	08	AE	9F	00058	PUSHAB	FAO_BUF	
	0C	AE	9F	0005B	PUSHAB	FAO_BUF	
	0000	CF	9F	0005E	PUSHAB	P.AAM	
00000000G	00	05	FB	00062	CALLS	#5, SYSSFAO	
	52	50	D0	00069	MOVL	R0, STATUS	
	05	52	E8	0006C	BLBS	STATUS, 3\$	0655
		52	DD	0006F	PUSHL	STATUS	
	64	01	FB	00071	CALLS	#1, LIB\$SIGNAL	
	03	63	F4	00074	SOBGEQ	LINE COUNTER, 4\$	0656
	66	00	FB	00077	CALLS	#0, PAGE HEADINGS	
DA	A3	6E	B0	0007A	MOVW	FAO_BUF, TRANS_RAB+34	
EO	A3	AE	D0	0007E	MOVL	FAO_BUF+4, TRANS_RAB+40	
	04	A3	9F	00083	PUSHAB	TRANS_RAB	
	B8	01	FB	00086	CALLS	#1, SYSSPUT	
65		50	D0	00089	MOVL	R0, STATUS	
52		52	E8	0008C	BLBS	STATUS, 5\$	
05		52	DD	0008F	PUSHL	STATUS	
	64	01	FB	00091	CALLS	#1, LIB\$SIGNAL	
	03	63	F4	00094	SOBGEQ	LINE COUNTER, 6\$	0657
	66	00	FB	00097	CALLS	#0, PAGE HEADINGS	
DA	A3	01	B0	0009A	MOVW	#1, TRANS_RAB+34	
EO	A3	CF	9E	0009E	MOVAB	P.AAO, TRANS_RAB+40	
	0000	A3	9F	000A4	PUSHAB	TRANS_RAB	
	B8	01	FB	000A7	CALLS	#1, SYSSPUT	
65		50	D0	000AA	MOVL	R0, STATUS	
52		52	E8	000AD	BLBS	STATUS, 7\$	
05		52	DD	000B0	PUSHL	STATUS	
	64	01	FB	000B2	CALLS	#1, LIB\$SIGNAL	
		04	000B5	7\$:	RET		0660

FILECDS
VJ4-000

FILECDS File-Oriented Phone Commands
PHN\$TRANSCRIBE_CMD - Handle TRANSCRIBE Command

B 1
16-Sep-1984 02:09:16
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742
[PHONE.SRC]FILECDS.B32;1

Page 14
(6)

; Routine Size: 182 bytes, Routine Base: \$CODE\$ + 0107

INF
V04

```
339 0661 2 ! We begin by opening a new file in which to place the transcript. We also
340 0662 2 ! have to connect the RAB so we can write records. If either of these actions
341 0663 2 ! fails, tell the user why.
342 0664 2
343 0665 2 trans_fab[fab$b_fns] = .spec_dsc[len];
344 0666 2 trans_fab[fab$l_fna] = .spec_dsc[ptr];
345 0667 2 status = $create(fab=trans_fab);
346 0668 2 if .status nequ rms$normal then (
347 0669 2     phn$inform(.status);
348 0670 2     return;
349 0671 2 );
350 0672 2 status = $connect(rab=trans_rab);
351 0673 2 if .status nequ rms$normal then (
352 0674 2     phn$inform(.status);
353 0675 2     $close(fab=trans_fab);
354 0676 2     return;
355 0677 2 );
356 0678 2
357 0679 2 ! Now we inform all of the people that we are going to record them.
358 0680 2
359 L 0681 2 phn$talk(describe(%string(%char(ret),
360 L 0682 2     'I AM NOW RECORDING THE CONVERSATION UP TO THIS POINT.',
361 0683 2     %char(ret))));
362 0684 2
363 0685 2 ! Now we are going to set up two arrays, containing entries for each person
364 0686 2 ! we are recording. The pub_array contain the address of the person's PUB.
365 0687 2 ! The ctl_array contains the address of the oldest CTL on their CTL chain.
366 0688 2
367 0689 2 ch$fill(0,%allocation(pub_array),pub_array);
368 0690 2 ch$fill(0,%allocation(ctl_array),ctl_array);
369 0691 2 p = .phn$gq_pubhead[0];
370 0692 2 i = 0;
371 0693 2
372 0694 2 until .p eqia phn$gq_pubhead do (
373 0695 2     if not .p[pub_v_temporary] and
374 0696 2         not .p[pub_v_uhaveheld] and
375 0697 2         not .p[pub_v_hasuheld] then (
376 0698 2
377 0699 2         pub_array[i] = .p;
378 0700 2         ctl_array[i] = .p[pub_q_ctlhead1];
379 0701 2         inc(i);
380 0702 2     );
381 0703 2     p = .p[pub_l_flink];
382 0704 2 );
383 0705 2
384 0706 2 ! Now we go into our merging loop. On each pass, we find the oldest stamp -
385 0707 2 ! that will merge the conversation text in the most accurate order. For
386 0708 2 ! each line we print one line in the transcript. It contains the conversion
387 0709 2 ! text and also the speaker's name unless it's the same speaker as the
388 0710 2 ! previous line.
389 0711 2
390 0712 2 page_number = 0;
391 0713 2 line_counter = 0;
392 0714 2 cur_p = 0;
393 0715 2
394 0716 2 loop (
395 0717 2
```



```
396 0718 3 ! Loop through arrays and find minimum stamp. If there are no
397 0719 3 ! CTLs left to merge, we're done.
398 0720 3
399 0721 3 min_stamp = -1;
400 0722 3 incfu i from 0 to array_size-1 do
401 0723 4 if .ctl_array[i] nequ 0 then (
402 0724 4 bind
403 0725 4 c = .ctl_array[i]: ctl;
404 0726 4
405 0727 5 if .c[ctl_l_stamp] lssu .min_stamp then (
406 0728 5 min_stamp = .c[ctl_l_stamp];
407 0729 5 min_i = .i;
408 0730 4 );
409 0731 4 );
410 0732 3
411 0733 3 exitif (.min_stamp eglu -1);
412 0734 3
413 0735 3 ! If this is a different person from the last line, then put the
414 0736 3 ! person's name on the beginning of the line. Otherwise just clear
415 0737 3 ! the beginning of the line.
416 0738 3
417 0739 4 if .pub_array[min_i] neqa .cur_p then (
418 0740 4 cur_p = .pub_array[min_i];
419 0741 4 transcribe(1,uplit byte(%char(ret)));
420 0742 4 begin
421 0743 5 bind
422 0744 5 new_tsb = cur_p[pub_b_tsb]: tsb;
423 0745 5 new_dsc = new_tsb[tsb_q_tkindsc,0]: descriptor;
424 0746 5
425 0747 5 ch$copy(.new_dsc[len],.new_dsc[ptr],' ',40,line_buf);
426 0748 4 end;
427 0749 4 ) else
428 0750 4 ch$fill(' ',40,line_buf);
429 0751 4
430 0752 4 ! Move the conversation text into the line.
431 0753 4
432 0754 4 c = .ctl_array[min_i];
433 0755 4 begin
434 0756 4 bind
435 0757 4 line_dsc = c[ctl_q_line]: descriptor;
436 0758 4
437 0759 4 ch$move(.line_dsc[len],.line_dsc[ptr],line_buf+40);
438 0760 4 transcribe(40+.line_dsc[len],line_buf);
439 0761 4 end;
440 0762 4
441 0763 4 ! Now we move to the previous CTL on this person's chain. If
442 0764 4 ! there isn't one, then zero the ctl_array entry.
443 0765 4
444 0766 4 ctl_array[min_i] = (if .c[ctl_l_blink] eglu cur_p[pub_q_ctlhead0]
445 0767 4 then 0
446 0768 4 else .c[ctl_l_blink]);
447 0769 3 );
448 0770 3
449 0771 2 ! All done. Close the transcription file and refresh the screen.
450 0772 2
451 0773 2 $close(fab=trans fab);
452 0774 2 phn$fresh_screen(false);
```

: 453
: 454
: 455

0775 2 return;
0776 2
0777 1 end;

```
52 4F 43 45 52 20 57 4F 4E 20 4D 41 20 49 0D 000A1 P.AAQ: .PSECT $SPLITS,NOWRT,NOEXE,2
52 45 56 4E 4F 43 20 45 48 54 20 47 4E 49 44 000B0 .ASCII <13>\I AM NOW RECORDING THE CONVERSATION\
4F 50 20 53 49 48 54 20 4F 54 20 50 55 20 4E 000C4 .ASCII \N UP TO THIS POINT.\<13>
OD 2E 54 4E 49 000D3
00000037 000D8 P.AAP: .LONG 55
00000000 000DC .ADDRESS P.AAQ
OD 000E0 P.AAR: .ASCII <13>

.EXTRN SYS$CREATE
.PSECT $CODE$,NOWRT,2
.OFFC 00000
.ENTRY PHN$TRANSCRIBE_CMD, Save R2,R3,R4,R5,R6,R7,-; 0603
MOVAB R8,R9,R10,R11
MOVL -180(SP), SP 0606
MOVAB TRANS_SPEC, R0 0665
MOVL (R0), TRANS_FAB+52 0666
MOVL 4(R0), TRANS_FAB+44 0667
PUSHAB TRANS_FAB
CALLS #1, SYS$CREATE
MOVL R0, STATUS
CMPL STATUS, #65537 0668
BEQL 1$
PUSHL STATUS 0669
CALLS #1, PHN$INFORM
RET 0668
PUSHAB TRANS_RAB 0672
CALLS #1, SYS$CONNECT
MOVL R0, STATUS
CMPL STATUS, #65537 0673
BEQL 2$
PUSHL STATUS 0674
CALLS #1, PHN$INFORM
PUSHAB TRANS_FAB 0675
CALLS #1, SYS$CLOSE
RET 0673
PUSHAB P.AAP 0683
CALLS #1, PHN$TALK
MOVCS #0, (SP), #0, #24, PUB_ARRAY 0689
MOVCS #0, (SP), #0, #24, CTL_ARRAY 0690
MOVL PHN$GQ_PUBHEAD, P 0691
CLRL I 0692
MOVAB PHN$GQ_PUBHEAD, R2 0694
CMPL P, R2
BEQL 5$
MOVAB 240(P), R2 0695
```

15	62	02	E0	0008C	BBS	#2, (R2), 4\$	
	12	62	E8	00090	BLBS	(R2), 4\$	0696
OE	62	01	E0	00093	BBS	#1, (R2), 4\$	0697
	E8 AD40	51	D0	00097	MOVL	P, PUB_ARRAY[I]	0699
	D0 AD40	0108	C1	D0 0009C	MOVL	264(P), CTL_ARRAY[I]	0700
	51	50	D6	000A3	INCL	I	0701
		61	D0	000A5	4\$: MOVL	(P), P	0703
		D3	11	000A8	3\$ BRB		0694
	0000'	CF	7C	000AA	5\$: CLRG	PAGE NUMBER	0712
	5A	57	D4	000AE	CLRL	CUR_P	0714
		01	CE	000B0	6\$: MNEGL	#1, MIN_STAMP	0721
	50	51	D4	000B3	CLRL	I	0722
	5A	0C	A0	D1 000BC	7\$: MOVL	CTL_ARRAY[I], R0	0723
	5A	0C	A0	D0 000C2	BEQL	8\$	0727
	58	51	D0	000C6	CMPL	12(R0), MIN_STAMP	
	05	51	D6	000C9	8\$: MOVL	12(R0), MIN_STAMP	0728
		51	D1	000CB	MOVL	I, MIN_I	0729
	FFFFFFF	8F	5A	D1 000D0	INCL	I	0723
		03	12	000D7	CMPL	I, #5	
	57	E8 AD48	D1	000DC	7\$ BLEQU		
		42	13	000E1	CMPL	MIN_STAMP, #-1	0733
	57	E8 AD48	D0	000E3	9\$ BNEQ		
	05	0000'	CF	F4 000E8	BRW	18\$	
	FE58	CF	00	FB 000ED	CMPL	PUB_ARRAY[MIN_I], CUR_P	0739
	0000'	CF	01	B0 000F2	BEQL	12\$	
	0000'	CF	01	B0 000F7	10\$: MOVL	PUB_ARRAY[MIN_I], CUR_P	0740
		0000'	CF	9E 000FE	SOBGEQ	LINE_COUNTER, -10\$	0741
	00000000G	00	01	FB 00102	CALLS	#0, PAGE HEADINGS	
	58	50	D0	00109	MOVW	#1, TRANS_RAB+34	
	09	5B	E8	0010C	MOVAB	P.AAR, TRANS_RAB+40	
	00000000G	00	01	FB 00111	PUSHAB	TRANS_RAB	
	50	10	A7	9E 00118	CALLS	#1, SYSSPUT	
28	20	04	B0	60 2C 0011C	MOVL	R0, STATUS	
			6E	00122	BLBS	STATUS, 11\$	
			06	11 00123	PUSHL	STATUS	
28	20	6E	00	2C 00125	12\$: CALLS	#1, LIB\$SIGNAL	
			6E	0012A	MOVAB	16(CUR_P), R0	0745
	56	D0 AD48	D0	0012B	13\$: MOVCS	(R0), #4(R0), #32, #40, LINE_BUF	0747
	59	10	A6	9E 00130	BRB	13\$	0739
28	AE	04	B9	69 28 00134	MOVCS	#0, (SP), #32, #40, LINE_BUF	0750
	05	0000'	CF	F4 0013A	MOVL	CTL_ARRAY[MIN_I], C	0754
	FE06	CF	00	FB 0013F	MOVAB	16(C), R9	0757
0000'	CF	69	28	A1 00144	14\$: MOVCS	(R9), #4(R9), LINE_BUF+40	0759
	0000'	CF	6E	9E 0014A	SOBGEQ	LINE_COUNTER, 14\$	0760
	00000000G	00	01	FB 00153	CALLS	#0, PAGE HEADINGS	
	58	50	D0	0015A	ADDW3	#40, (R9), TRANS_RAB+34	
	09	5B	E8	0015D	MOVAB	LINE_BUF, TRANS_RAB+40	
	00000000G	00	01	FB 00162	PUSHAB	TRANS_RAB	
		50	D0	00169	15\$: CALLS	#1, SYSSPUT	
		5B	DD	00160	MOVL	R0, STATUS	
		01	FB	00162	BLBS	STATUS, 15\$	
		C7	9E	00169	PUSHL	STATUS	
					CALLS	#1, LIB\$SIGNAL	
					MOVAB	260(R7), R0	0766

FILECMDS
V04-000

FILECMDS File-Oriented Phone Commands
PHN\$TRANSCRIBE_CMD - Handle TRANSCRIBE Command

G 1
16-Sep-1984 02:09:16
14-Sep-1984 12:53:25

VAX-11 Bliss-32 V4.0-742
[PHONE.SRC]FILECMDS.B32;1

Page 19
(7)

```

      50      04      A6      D1      0016E      CMPL      4(C), R0
      04      12      00172      BNEQ      16$
      50      D4      00174      CLRL      R0
      04      11      00176      BRB      17$
      50      04      A6      D0      00178      16$:      MOVL      4(C), R0
DO AD48      50      D0      0017C      17$:      MOVL      R0, CTL_ARRAY[MIN_I]
      FF2C      31      00181      BRW      6$
      0000'      CF      9F      00184      18$:      PUSHAB     TRANS_FAB
      00000000G      00      01      FB      00188      CALLS      #1, SYSSCLOSE
      0000G      CF      7E      D4      0018F      CLRL      -(SP)
      01      FB      00191      CALLS      #1, PHN$FRESH_SCREEN
      04      00196      RET

```

0768
0766
0714
0773
0774
0777

: Routine Size: 407 bytes, Routine Base: \$CODE\$ + 0:BD

: 456 0778 1
: 457 0779 0 end eludom

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	384 NOVEC, WRT, RD	,NOEXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)
\$PLITS	225 NOVEC,NOWRT, RD	,NOEXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)
\$CODE\$	852 NOVEC,NOWRT, RD	, EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	42	0	581	00:00.8

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:FILECMDS/OBJ=OBJ\$:FILECMDS MSRC\$:FILECMDS/UPDATE=(ENH\$:FILECMDS)

: Size: 852 code + 609 data bytes
: Run Time: 00:14.2
: Elapsed Time: 01:06.3
: Lines/CPU Min: 3291
: Lexemes/CPU-Min: 43364

FILECMDS
V04-000

FILECMDS File-Oriented Phone Commands
PHN\$TRANSCRIBE_CMD - Handle TRANSCRIBE Command

H 1
16-Sep-1984 02:09:16

VAX-11 Bliss-32 V4.0-742

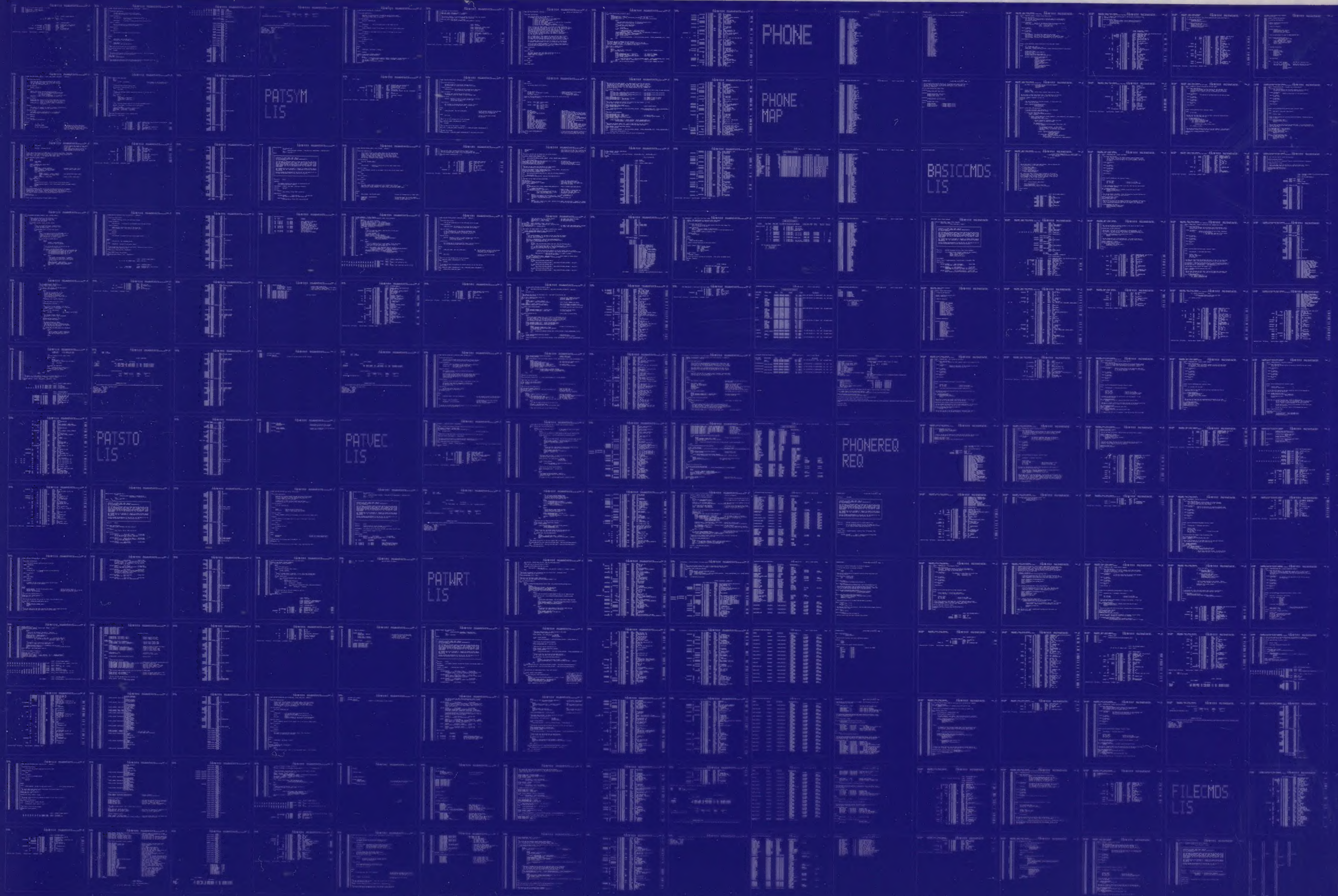
Page 20

: Memory Used: 190 pages
: Compilation Complete

INI
VO

0304 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0305 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY